Selection in Scale-Free Small World

Zsolt Palotai[1], Csilla Farkas[2], and András Lőrincz[1,⋆]

[1] Eötvös Loránd University, Department of Information Systems,
Pázmány Péter sétány 1/c, Budapest, H-1117, Hungary
andras.lorincz@elte.hu

[2] University of South Carolina, Department of Computer Sciences and Engineering,
Columbia, SC 29208, USA

Abstract. In this paper we compare our selection based learning algorithm with the reinforcement learning algorithm in Web crawlers. The task of the crawlers is to find new information on the Web. We performed simulations based on data collected from the Web. The collected portion of the Web is typical and exhibits scale-free small world (SFSW) structure. We have found that on this SFSW, the weblog update algorithm performs better than the reinforcement learning algorithm. It finds the new information faster than the reinforcement learning algorithm and has better new information/all submitted documents ratio.

1 Introduction

The largest source of information today is the World Wide Web. The ever-increasing growth of the Web presents a considerable challenge in finding novel information on the Web. In addition, properties of the Web, like scale-free small world (SFSW) structure [1,2] may create additional challenges. For example the direct consequence of the scale-free small world property is that there are numerous URLs or sets of interlinked URLs, which have a large number of incoming links. Intelligent web crawlers can be easily trapped at the neighborhood of such junctions as it has been shown previously [3,4].

In this paper we present a selection based algorithm and compare it to the well-known reinforcement learning algorithm in terms of their efficiency and behavior. The selection algorithm, called weblog update, modifies the starting URL lists of our crawlers based on the found relevant documents. The reinforcement learning algorithm modifies the URL orderings of the crawlers based on the received reinforcements for submitted documents. We have found that the weblog update selection algorithm performs better in this environment than the reinforcement learning algorithm, eventhough the reinforcement learning algorithm has been shown to be efficient in finding relevant information [4,5].

The paper is organized as follows. We overview the forager architecture in Section 2. After that in Section 3 we present our experiment on the Web and the conducted simulations with the results. Section 4 concludes our paper.

[⋆] Corresponding author.

M. Pěchouček, P. Petta, and L.Z. Varga (Eds.): CEEMAS 2005, LNAI 3690, pp. 579–582, 2005.

2 Forager Architecture

There are two different kinds of agents: the foragers and the reinforcing agent (RA). The fleet of foragers crawl the web and send the URLs of the selected documents to the reinforcing agent. The RA determines which forager should work for the RA and how long a forager should work. The RA sends reinforcements to the foragers based on the received URLs.

Foragers may use two different kinds of algorithms to find relevant documents. The first algorithm, called weblog update algorithm selects the possibly good starting URLs and restarts periodically the forager from one of the possibly good starting URLs. The second algorithm is the reinforcement learning based URL ordering algorithm which selects the next document to be visited by the forager between two restarts. The documents found by a forager are represented as a 50 dimensional state vector. This algorithm updates the 50 dimensional weight vectors of the foragers in order to collect more rewards in the long run for sending relevant documents to the RA. The document to be visited next by the forager is the one with the highest value, where the value of a document is the scalar product of its state vector and the forager's weight vector. According to the weblog update algorithm the starting URL value of an URL is the sum of rewards collected during steps after visiting that URL and before the next restart. URLs with low starting URL values fall out from the weblog, while URLs of documents with high starting URL values go to the front of the weblog.

The algorithms and the architecture are detailed in [6].

3 Experiments

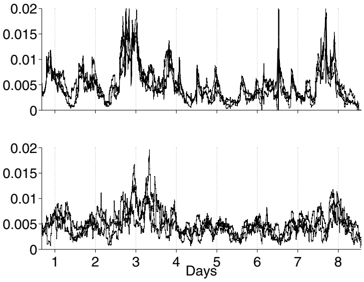
We conducted an 18 day long experiment on the Web to gather realistic data. We used the gathered data in simulations to compare the weblog update and reinforcement learning algorithms.

In the web experiment a fixed number of foragers were competing with each other to collect news at the CNN web site. The foragers were running in equal time intervals in a predefined order on one PC. We deployed 8 foragers using the weblog update and the reinforcement learning based URL ordering update algorithms. We also deployed 8 other foragers using the weblog update algorithm but without reinforcement learning. We used this heterogenous fleet of foragers to eliminate any biases on the gathered data. We investigated the link structure of the gathered Web pages. We have found that the links have a power-law distribution. That is the link structure has the scale-free property. We have also found that the links of gathered pages form small world structure.

We conducted simulations with two different kinds of foragers. The first case is when foragers used only the weblog update algorithm without URL ordering update (WL foragers). The second case is when foragers used only the reinforcement learning based URL ordering update algorithm without the weblog update algorithm (RL foragers). The simulation for each type of foragers were repeated 3 times with different initial weight vectors for each forager.

type	RL	std RL	WL	std WL
downloaded	540636	9840	669673	9580
sent	9747	98	6345	385
relevant	2419	45	3107	60
found URLs	31092	1050	33116	3370
download eff	0.0045	0.0001	0.0046	0.0001
sent eff	0.25	0.003	0.49	0.031
exploration	0.058	0.001	0.050	0.006
freshness	0.70	0.006	0.74	0.011
age (hours)	1.79	0.04	1.56	0.08

(a) Simulation results



(b) Efficiency

Fig. 1. Simulation results and Efficiency. **(a):** 2^{nd} (3^{rd}) and 4^{th} (5^{th}) columns show averages (standard deviations) of individual experiments. **(b):** Horizontal axis: time in days. Vertical axis: download efficiency, that is the number of found relevant documents divided by number of downloaded documents in 3 hour time intervals. Upper subfigure shows RL foragers' efficiencies, lower subfigure shows WL foragers' efficiencies. For all of the 3 simulation experiments there is a separate line.

Table 1(a) in Fig. 1 shows the measured parameter values averaged over the 3 runs of each type of foragers. From Table 1(a) we can conclude the followings. The efficiencies of RL and WL foragers from the point of view of the news site are about the same (download efficiency). From the point of view of the RA the efficiency of WL foragers is higher than RL foragers (sent efficiency). This shows that WL foragers divide the search area better among each other than RL foragers. Sent efficiency would be 1 if none of two foragers have sent the same document to the RA. RL foragers explore more than WL foragers: RL found more URLs per downloaded page than WL foragers did (exploration). WL foragers find faster the new relevant documents in the already found clusters. That is freshness is higher and age is lower than in the case of RL foragers.

Fig. 1(b) shows other aspects of the different behaviors of RL and WL foragers. Download efficiency of RL foragers has more, higher, and sharper peaks than the download efficiency of WL foragers has. That is WL foragers are more balanced in finding new relevant documents than RL foragers. The reason is that while the WL foragers remain in the found good clusters, the RL foragers continuously explore the new promising territories. The sharp peaks in the efficiency show that RL foragers *find and recognize* new good territories and then *quickly collect* the current relevant documents from there.

4 Conclusions

We presented and compared our selection algorithm to the well-known reinforcement learning algorithm. Our comparison was based on finding new relevant documents on the Web, that is in a dynamic scale-free small world environment. We have found that the weblog update selection algorithm performs better in this environment than the reinforcement learning algorithm, eventhough the reinforcement learning algorithm has been shown to be efficient in finding relevant information [4,5]. We explain our results based on the different behaviors of the algorithms. That is the weblog update algorithm finds the good relevant document sources and remains at these regions until better places are found by chance. Individuals using this selection algorithm are able to quickly collect the new relevant documents from the already known places because they monitor these places continuously. The reinforcement learning algorithm explores new territories for relevant documents and if it finds a good place then it collects the existing relevant documents from there by quickly adapting to the new neighborhood. Although RL is more flexible and has a fast tuning mechanims, nevertheless RL finds new relevant documents slower on the average than the more conservative weblog update algorithm. We conjecture that this conclusion may be restricted to highly clustered worlds, e.g., to scale-free small worlds.

Acknowledgement

This material is based upon work supported by the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory, under Contract No. FA8655-03-1-3036. This work is also supported by the National Science Foundation under grants No. INT-0304904 and No. IIS-0237782. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory.

References

1. Barabási, A., Albert, R., Jeong, H.: Scale-free characteristics of random networks: The topology of the world wide web. Physica A **281** (2000) 69–77
2. Kleinberg, J., Lawrence, S.: The structure of the web. Science **294** (2001) 1849–1850
3. Kókai, I., Lőrincz, A.: Fast adapting value estimation based hybrid architecture for searching the world-wide web. Applied Soft Computing **2** (2002) 11–23
4. Lőrincz, A., Kókai, I., Meretei, A.: Intelligent high-performance crawlers used to reveal topic-specific structure of the WWW. Int. J. Founds. Comp. Sci. **13** (2002) 477–495
5. Rennie, J., Nigam, K., McCallum, A.: Using reinforcement learning to spider the web efficiently. In: Proc. 16th Int. Conf. on Machine Learning (ICML), Morgan Kaufmann, San Francisco (1999) 335–343
6. Palotai, Z., Farkas, C., Lőrincz, A.: Selection in scale free small world. http://www.arxiv.org/pdf/cs.LG/0504063 (2005)